IceCube Software System Design Methodology

DAQ S/W Workshop

28 Oct 2002

John Cavin

IceCube Software Vision

- “To provide the physicist, in collaboration with the DAQ system, with stable, reliable,reproducible access to data taken by the IceCube detector and data created by any approved simulation of the detector, so that they can publish physics papers.”
  - Icecube Software Design Description

Major IceCube Software Components

- Production
  - DAQ
  - Data Handling
  - Data Warehouse
  - Analysis Framework
- Simulation
- Analysis

Software Design Approach

- Create an architecture that encourages the development of software components in parallel
  - Partition into loosely coupled entities
  - Encapsulate (Hide the details)
  - Define clear interfaces

IceCube Software Principles

- Open development
- Common coding standards
- Write Tests before code
- Daily builds
- Automated build process with integrated configuration management
- Continuous integration
- Regression testing
- Frequent releases
- Maintainable

Software Design Observations

- Design is not a linear process
- It is often messy and iterative
- Highly dependent on people’s experience
- Abstraction helps prevent getting trapped in the details
- Typically not a single optimum solution

IceCube S/W Process

- Started with:
  - LBNL system architecture proposal (PDD)
  - Requirements for system (SRS)
  - Existing String 18 software
  - AMANDA simulations
  - AMANDA analysis (Siegmund)

DAQ Software Process

- DAQ Systems Requirements
- DAQ MainBoard Hardware Requirements
- DAQ MainBoard Software Requirements
- DAQ Software Description Document

DAQ Major S/W Components

- DOM MB Software
- DOM Hub Software
- String Processors
- Trigger Processors
- Global Trigger Processor
- Event Builder

Budget Limitations

- DAQ Software efforts primarily limited this budget to:
  - DOM MB Software
  - DOM Hub Software
  - DOM Testing Software

IceCube’s Problems (opportunities)

- How do we convert this information into a good design that meets the needs of the physicists and captures the vision of IceCube software?
- How do we incorporate the talent of the collaboration into a successful software development?

IceCube S/W Approach

- Provide a set of tools (development environment) that enables collaborators to use the IceCube software principles
- Partition system into subsystems with emphasis on defining entities and interfaces that allow maximum independence
- Document this top level architecture in Software Description Document (SDD) and review
- Create a top level design for the subsystems (SDD) and review
- Have responsible collaborators create detailed designs for subsystem components and review with subsystem managers

Development Environment

- An integrated set of tools that makes it easier for the developer to code, test, build, integrate software continuously:
  - CVS
  - Junit
  - ANT
  - Cruise Control
  - TenderBox
  - Issuezilla

What is a Software Design Description?

- Provides following views of the system:
  - Decomposition Description (defines entities)
  - Dependency Description (relationships among entities)
  - Interface Description (external view of entities)
  - Detail Description (internal view of entities)
- Based on IEEE standard
- See Example documents

IceCube Software Design Attributes

- Loose Coupling
- High Cohesion
- Encapsulation

- Enables parallel development

What Next?

- Need to identify collaborators who are interested in taking responsibility for portions of IceCube software
  - Want to work to in an open, collaborative environment
  - Willing to help improve the process
  - Willing to work as a team

Software Responsibilities

- System Architecture – Simon Patton
- DAQ Software – Chuck McParland
- Data System – Bill Carithers
- Simulation – Greg Sullivan
- Analysis – Doug Cowen
- Software Coordinator – John Cavin